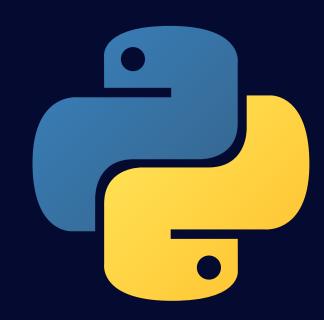
GENERADOR DE CONTRASEÑAS SEGURAS EN PYTHON

Josué Paucar



Propósito del Proyecto

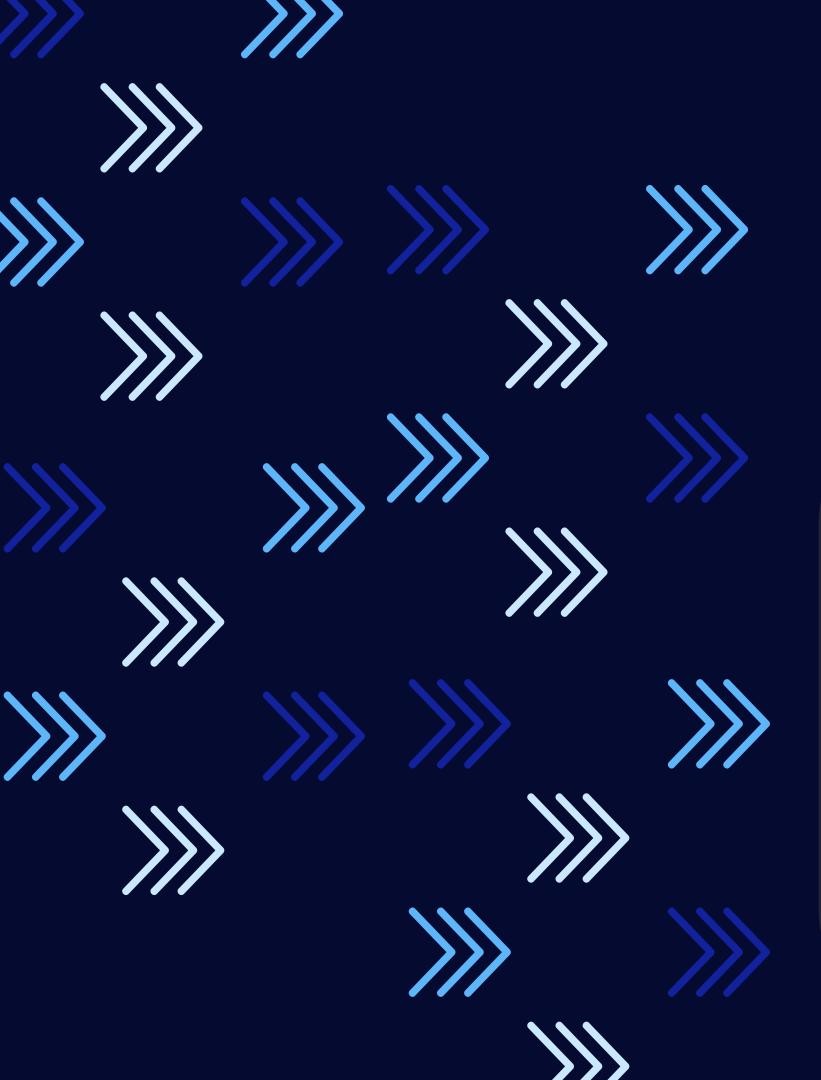
Desarrollar una herramienta para generar contraseñas seguras que protejan datos personales y eviten accesos no autorizado.

Importancia de la Seguridad

Las contraseñas seguras previenen el robo de identidad y accesos indebidos, protegiendo la confidencialidad de los datos.

Objetivos del Programa

- Generación Automática: Crear contraseñas con letras, números y símbolos.
- Facilidad de Uso: Interfaz simple y eficiente.
- Validación de Seguridad: Verificaciones de seguridad integradas.
- Adaptabilidad: Permite ajustar la longitud de la contraseña.



DESCRIPCIÓN DEL CÓDIGO

Importación de Librerías y Definición de Variables

- Importación de librerías necesarias: random y string
- Definición de variables clave: bandera1 y longitud

```
1 # Importación de Librerías
```

- 2 import random
- 3 import string

4

5 # Definición de Variables

6 bandera1 = True

7 longitud = int

8



SOLICITUD DE LONGITUD DE CONTRASEÑA

Solicitud y Validación de la Longitud de la Contraseña

- Implementación del bucle while para validar la longitud
- Uso de try y except para manejar excepciones

```
while bandera1 == True: # Se utiliza una bandera para cuando el
     ingreso del dato sea correcto salga del bucle y continue con el
     resto del código
         try: # Se utiliza try y except para manejar excepciones en
14
         caso de que el usuario pueda introducir un dato no solicitado
             dato = input("Ingrese la longitud deseada de la
15
             contraseña a generar, en valor numérico entero del 8 -
             30: ")
             longitud = int(dato) # En caso de no poder convertir
16
             este dato a entero, se manejará con el except
             if 8 <= longitud <= 30: # Se verifica que el ingreso
17
             del entero sea entre los valores de 8 a 30
                 bandera1 = False # En el caso de que el dato
18
                 ingresado sea correcto, la bandera se apaga y esto
                 permite salir del bucle
19
             else:
                 print("Por favor ingrese un número entre 8 y 30.")
20
         except ValueError:
21
             print("¡No se ha ingresado un caracter numérico entero!")
22
```



CONJUNTO DE CARACTERES

Definición del Conjunto de Caracteres

- Uso de string.ascii_letters, string.digits y string.punctuation
- Importancia de incluir letras, dígitos y símbolos

```
# Conjunto de caracteres disponibles para generar la contraseña
caracteres = string.ascii_letters + string.digits + string.
punctuation # Conjunto de caracteres (letras, dígitos y
símbolos) que se utilizarán
```



GENERACIÓN DE LA CONTRASEÑA

Generación de la Contraseña

- Se genera la contraseña utilizando una lista de compresión
- Hay un proceso de verificación de la contraseña para asegurar la inclusión de letras, dígitos y símbolos

```
# Bucle para generar y verificar la contraseña
while True:
    # Uso de la lista de compresión para crear la contraseña
   contraseña = ''.join(random.choice(caracteres) for i in range(longitud)) # Generación de la contraseña
   # Verificación de que la contraseña tenga al menos una letra, un dígito y un símbolo
   tiene letra = False
   tiene_digito = False
   tiene simbolo = False
   for c in contraseña:
       if c.isalpha(): # Verificar si el carácter es una letra
            tiene letra = True
       if c.isdigit(): # Verificar si el carácter es un dígito
            tiene digito = True
        if c in string.punctuation: # Verificar si el carácter es un símbolo
            tiene simbolo = True
   # Si la contraseña cumple con los 3 requisitos, salir del bucle
   if tiene_letra and tiene_digito and tiene_simbolo:
       break # La contraseña es válida, salimos del bucle
   else:
       print("La contraseña generada no cumple con los requisitos de tener letras, números y símbolos. Generando una nueva...")
```



VERIFICACIÓN DE LA CONTRASEÑA

Verificación de la Contraseña Generada

- Bucle for para verificar la contraseña generada
- Criterios de verificación: al menor una letra, un dígito y un símbolo

```
for c in contraseña:

if c.isalpha(): # Verificar si el carácter es una letra

tiene_letra = True

if c.isdigit(): # Verificar si el carácter es un dígito

tiene_digito = True

if c in string.punctuation: # Verificar si el carácter es un símbolo

tiene_simbolo = True

# Si La contraseña cumple con Los 3 requisitos, salir del bucle

if tiene_letra and tiene_digito and tiene_simbolo:

break # La contraseña es válida, salimos del bucle

else:

print("La contraseña generada no cumple con los requisitos de tener letras, números y símbolos. Generando una nueva...")
```



FIN DEL PROGRAMA

Resultado del Programa

- Presentación de la contraseña generada al usuario.
- Finalización del programa

